Server Backups And Restores

Your organization can be prepared to recover from data loss by performing proper backup planning and implementation. Developing a backup and restore strategy, carrying out that strategy and maintaining database consistency can improve the integrity of any Exchange 2000 database. It is important to understand the Exchange 2000 Database technology as well as the different categories of backup that are available. Knowing the prerequisites and processes involved will help you successfully backup data to prevent data loss and restore data when needed.

In this Chapter:

Microsoft Exchange 2000 Server Database Technology

Backup Prerequisites

Backup Categories

Backup Process

Restore Process Overview

Prerequisites for Restoring Data

Restore Categories

Database Consistency

Recovering from Disasters

Best Practices

Microsoft Exchange 2000 Server Database Technology

> **Important** Before you back up or restore your Exchange Server data, you should download the latest hot fix from the Microsoft 2000 Exchange web site at http://www.Microsoft.com/exchange to get the most recent version of NTBackup.exe.

Understanding the underlying database technology that Exchange 2000 uses can help you to better understand the backup and restore process. The database technology is based on the Extensible Storage Engine. This is part of the Web Storage System process. Users of the affected database will not be able to access their mail during a restore. The Extensible Storage Engine also has features that allow Exchange 2000 to service users while the server is backed up. Figure X.X. shows a current database.

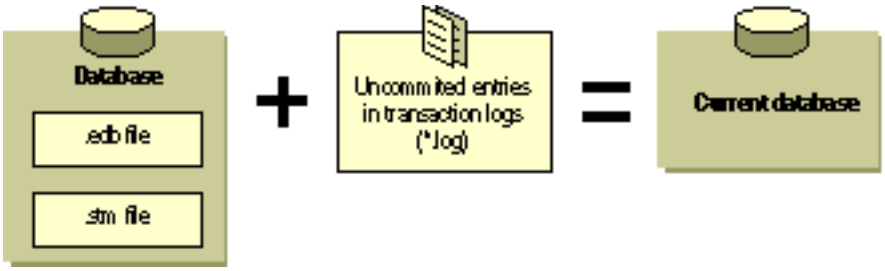

Figure x.x Description of a Current Database

Exchange 2000 uses Windows 2000 Active Directory. In previous versions of Exchange, the directory is an ESE database that is backed up with other Exchange databases. Because the Exchange 2000 directory is part of Windows, backing up Windows 2000 Active Directory is as important as backing up Exchange 2000 databases.

Extensible Storage Engine Structure

The Extensible Storage Engine uses a balanced-tree (B-tree) structure to store data. Each page in the database file is a node in the B-tree structure. An Extensible Storage Engine database can contain up to 2^32 pages or 16 terabytes (Active Directory uses 8 Kilobyte (KB) pages and can contain up to 32 terabytes). This means that your database size is limited only by your ability to back up and restore the database in a timely way.

Multiple Storage Groups and Failover

An Exchange storage group is a part of the Extensible Storage Engine. Each storage group consists of the Properties Database (edb) files, Streaming Database (stm) files and the set of log files for all the databases in the storage group. Exchange 2000 Server provides multiple storage groups and multiple databases within each storage group. Failover is an event where the active node stops functioning, or a critical service on the active node fails, and the secondary node takes its place. When you install Microsoft Exchange 2000 Server on each node, you use the Cluster Administrator to define the failover rules for Exchange 2000 services. These rules define the conditions that trigger a failover to the secondary node. The node that takes over must have access to the data at the moment of failure, so that users and jobs have current data.

Databases

You can create multiple databases within each storage group and restore databases in a storage group individually or together. A single set of log files contains entries for all databases in the storage group. One database within the storage group is composed of the following files:

- The .edb file, which contains folders, tables and indexes for messaging data; as well as MAPI messages and attachments.
- The .stm file is a new format in Exchange 2000 Server for storing native Internet content. The internal schema for the stm pages is stored in the .edb file. The .edb file stores messages in rich-text format. The .edb and .stm files function as a pair, and the database signature (a 32-bit random number combined with the time the database was created) is stored as a header in both files.

- Site Replication Service (SRS) files permit compatibility Exchange 5.5 by emulating an Exchange 5.5 directory service.
- Key Management Server (KMS) provides security encryption services.

Both the .edb and .stm files are organized into 4KB pages. In the .edb file the pages are organized into a B-tree. However, in the .stm file the pages are grouped in a clustered block or run format, similar to a file system such as NTFS. These pages are stored in 64 KB blocks (16 runs of 4kb each) within the .stm file on the hard disk.

Exchange data is stored in .edb, .stm, .srs and .kms files. These are the database files that share the same attributes as the operating system files. .edb files are organized into a collection of pages that store data definitions, data and indexes. These pages are numbered sequentially in a database file.

Logs form in the following way:

1. When the current log file fills, the edbtemp.log file starts.
2. The original log file name changes to the true generation number, so that it has the edb prefix followed by a five-digit hexadecimal number.If the first generation file name is edb000A1.log, the next edb.log file name is edb000A2.log.
3. The file edbtemp.log becomes edb.log, and the process repeats.

   Note Exchange 2000 uses the base name, not the edb file name prefix. They are presented in this format for purposes of clarity. The actual file base names appear differently in your directory.

Database GUID

A database GUID (Global Unique Identifier) is assigned to the store database, and a matching GUID is stored in Active Directory. This is very important, because the database does not mount if the GUIDs do not match.

Mailbox GUID

Each mailbox in the database has a GUID. The user account in Active Directory has the GUID of the mailbox which it owns. This means that even though you can delete a user from Active Directory, the mailbox still exists in the database. Therefore, you can reconnect the same mailbox to a different user. If the user account is accidentally deleted, you can be fix it by reconnecting the mailbox. It adds the mailbox GUID as a property field to the user account in Active Directory. The mailbox itself has a default retention time of 30 days. For more information about how to recover a single mailbox, see "Restoring a Single Mailbox", later in this chapter.

Transaction Log Files

Changes to the database are also stored in transaction log files. These files store a sequential list of every operation that is performed on a page in memory. These log files give you a way to restore committed transactions that you can lose when there is a power failure or some other type of disaster. Exchange 2000 Server log files are always 5 Megabytes (MB) in size. If you view them in Windows Explorer, they show up as 5,242,880 bytes. If the log files do not show this exact size, typically they are damaged.

Each storage group service that uses the Extensible Storage Engine (ESE) reserves two log files: Res1.log and Res2.log. These files are stored in the log directory. Reserve log files are used as placeholders for extra disk space. However, they are the same size as normal log files, and they function like normal log files. They are used if the service runs out of space. It is recommended that you store log files and database files on separate drives.

Log File Signature

You can use the log file signature to your advantage. If you shut down the server and inadvertently delete all log files, when you restart the server the Extensible Storage Engine creates a new series of log files, starting with a generation number of one. Because log files can have the same name, the Extensible Storage Engine stamps the header in each file series with a unique signature so it can distinguish between different series of log files.

Checkpoint File

Checkpoint files store Information that indicates when a transaction successfully saves to the hard disk. The Edb.chk file points to the log file that has had all transactions successfully committed to the database file. When all the pages that are changed by transactions in a given logfile write to the database file, the checkpoint advances to the log file with the next unwritten entry. Separate Edb.chk files are maintained for each storage group. The Edb.chk file is not required if you want to replay transactions. The Extensible Storage Engine determines which transactions are already written. It does this by examining the transaction log files. However, using the checkpoint saves the Extensible Storage Engine from starting at the first log file and checking every operation.

Figure X.X shows a description of Checkpoint files.

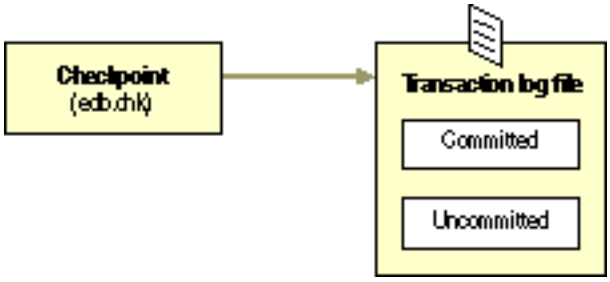

Figure X.X Description of Checkpoint Files

Circular Logging

Circular Logging is a process that reduces disk storage requirements by overwriting transaction logs after transactions are committed to the databases. Circular Logging is used with non-critical data, and it is disabled by default. Transactions to the .stm files are not logged. This is because the data is saved into the .stm file before the transaction commits. Therefore less data is written, which increases the speed of the process and frees resources. When Circular Logging runs, changes to the .edb file are written to the transaction logs. In a restore operation, you can play back data only to the point of the last backup. These transactions are entered into the restored database file to bring the

database up-to-date to the point in time where the last backup was performed, and not to the point where the database files went offline. It is recommended that you keep circular logging disabled because you can lose information during a restore operation.

Checksum

The Checksum (also called a message hash) is a string of bits 4 bytes in size that is calculated and then added to the page. Every Exchange Database (.edb) File is made up of pages that are 4 kilobytes (kb) in size. The integrity of these pages is checked through a checksum and also a page number (4 bytes in size) in the header of the database page. The first 82 bytes of the database page contain the header information. The header also contains flags for the type of page and information about what kind of data the page contains. When Exchange 2000 reads pages out of the database, these pages are then compared for the correct page number, as well as the checksum, which is calculated to ensure that the data in the page is not damaged. If the data is damaged, then the ESE returns an error to the Web Storage System, the database is stopped, and an event is logged informing the administrator of damage and actions to take (completing a restore for example). This ensures the optimal integrity of the database by informing you immediately if your database is damaged.

Individual Database Backup

In Exchange 2000,you use the NTBACKUP tool included with Windows 2000 to perform an individual backup. It is recommended that the you download the latest hotfix or service pack which includes this tool. NTBackup recognizes multiple storage groups and multiple databases. The level of detail for backup and restore is determined at the database level. It is possible to restore an individual database inside a storage group without restoring an entire storage group. If you tag an individual database for backup, it backs up the .edb file and the .stm file as well as the needed transaction log files. It is recommended that, as part of your backup strategy, you tag an entire storage group for backup. This is because if you backup databases individually, in the process you will also backup log files multiple times.

Exchange 2000 Server Clusters

Exchange 2000 Server supports active/active clustering. Clustering is managing a group of independent servers as a single system. These servers have individual memory, processors, and Network Interface cards and a shared storage medium (for example, Redundant Array of Independent Drives, or RAID. The cluster of servers must have identical processors and the same amount of RAM. You connect these computers using a network cable that puts them in continuous contact with each other. If a server fails, the other members of the cluster take over the failed server’s clients. There may be a short period of time in which the clients are unable to connect to the network, but when the hand-off is completed normal working conditions resume.

A cluster consists of two or more physical computers (nodes) connected to each other and a shared storage device. One or more Exchange virtual servers exist in the cluster, and each virtual server runs on one of the nodes in the cluster. Exchange can support multiple virtual servers on a single node. Clients connect to the virtual servers the same way that they would connect to a stand-alone server.

The limit to the number of storage groups found on a single Exchange 2000 server is 4 storage groups with 5 databases per storage group. Each storage group has a temporary database containing temporary tables and is used for large sorts. Exchange also holds 12 storage groups in reserve for restore. For a server cluster the web storage system can mount 4 storage groups. If one of the nodes stops functioning, the other node must be able to support the storage groups of both nodes. Figure X.X. shows the files found in a Storage Group.1

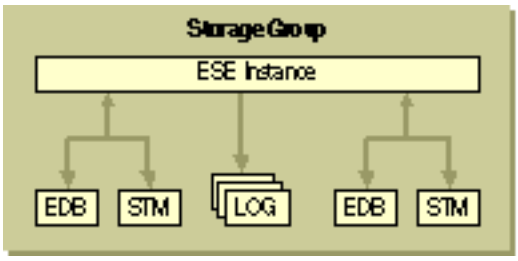

Figure x.x Storage Group Files

The unit of failover is the Exchange cluster group (also called the Exchange Virtual Server). Exchange does not take over in a failover event. The cluster service takes over in a failover event. Clustering resources, such as MSExchangeIS and MSExchangeMTA, are grouped together into a cluster group through the cluster administrator. When one of these resources fails, then the entire group fails because these resources depend on each other. Then the cluster service moves the group to another node in the cluster. The cluster group, or Exchange Virtual Server is a unit of failover. There might be a brief interruption in service during a failover event, although clients such as Outlook usually reconnect automatically.

Exchange Clustering provides high availability for messaging. The cluster service monitors the virtual servers in the cluster. In the event of a failure, the cluster service restarts or moves the affected virtual servers to a healthy node. For planned outages, you can manually move the virtual servers to other nodes. In either event, the client experiences an interruption of service only during the brief time that the virtual server is offline. The ability to use multiple servers at all times reduces system costs while increasing reliability, since no dedicated fail-over only, or hot stand-by servers are needed.

Clustering support differs according to the component. Table X.X details which components are supported, and in some cases, the type of clustering they are capable of supporting.

In table X.x the following terms are used to describe the level of functionality that each component supports.

- Active/passive: only one instance of the component can run in a cluster.
- Active/active: multiple instances of the component can run in a cluster.

Table X.X Clustering support

Components	Notes	Cluster Functionality

Exchange System Attendant		active/active
Web Storage System	After failover: each node can support 4 Storage Groups (with more in reserve).	active/active
Message Transfer Agent (MTA)	One instance per cluster. The MTA will be in only one Exchange Virtual Server.	active/passive
POP3	Multiple virtual servers per node.	active/active
IMAP	Multiple virtual servers per node.	active/active
SMTP	Multiple virtual servers per node.	active/active
Chat		active/passive
MSSearch		active/active

Backup Prerequisites

To successfully back up the data on your Exchange 2000-based servers, you must satisfy the necessary prerequisites. You need to define a backup strategy, fulfill the resource requirements, have the proper Web Storage System, define the data you need to back up, verify the integrity of your configuration data and address performance considerations.

Planning a Backup Strategy

Your backup strategy determines your restore strategy. These operations cannot be planned separately. When you create a backup strategy, you should also consider the time it takes to restore. Make sure you have enough hard disk drive capacity to restore both the database and the log files. For example, a weekly full backup plus one week of transactional log files might be more than your server can store, depending on how many log files are generated during a week. If your server generates 2000 log files in a week, this amounts to 10 gigabytes of log file space, in addition to the database.

Creating and Verifying Daily Backups

Creating and verifying daily backups is a critical step for successful restores. You can verify with practice restores and then examine the detailed log files using NTBackup. However, you can only restore valid data if the backup is valid. Failure to verify backups is one of the most common mistakes you might make because it is easy to assume that backup tapes are swapped and that data is backed up properly. Make it part of your daily routine to review all backup logs, and follow up on any errors or inconsistencies.

Standardizing Tape Backup Formats

Restore equipment must be compatible with production tape equipment. If you deploy a new type of tape drive, make sure that you use a compatible model for your restore equipment. You should also test reading and restoring production tape backups on the tape drive you use for restore.

Publishing a Maintenance Schedule

It is important to set user expectation levels by publishing a maintenance schedule, especially when users expect service 24 hours a day, seven days a week. Maintenance is inevitable because of the nature of data processing upgrades.

Checking for Hot Fixes

Microsoft releases hot fixes to correct problems that might occur. Therefore check Microsoft's FTP site on a monthly basis for the latest release of hot fixes. Before you install a hot fix, run it on a test server first. Otherwise it may cause serious disruptions over your network. As a rule of thumb, always utilize hot fixes related to security, but only implement other hot fixes if you are experiencing the specific problems for which the hot fix was designed.

Deciding Types of Data to Back Up

It is important to backup mission critical data, which is typically the user's data. This might include configuration information, mailboxes, or databases. It depends on your organization's priorities. You should always have a full backup of your static data, and routinely backup your dynamic data. Both types of data are described here:

Static Data

Static data includes the following types of information:

- Windows 2000 operating system and any service packs or hot fixes
- Packaged application software (Exchange)
- Supporting software, such as third-party backup software, or system management software
- User application software, such as ASP applications, mailbox agents, and workflow software
- Management scripts

Dynamic Data

Dynamic data includes the following types of data:

- Exchange Web Storage System databases and supporting files
- Message tracking log
- Active Directory
- Key Management Service(KMS) databases (needs to run on a member server with certificate authority)

- Site Replication Service (SRS) databases
- System State including IIS metabase (and quorom if on a cluster)

  Note The composition of System State varies, depending on whether you use Windows 2000 Professional or Windows 2000 Server. Both versions of Windows 2000 have System States that include: Boot files, COM+ Class Registration database and the Registry. But Windows 2000 Server System State also includes: Active Directory (if the server is a domain controller), Sysvol (if the server is a domain controller), Certificate Services database (if this is run), and Quorom (if the server is part of a cluster).

Resource Requirements

Before you back up your server, you must ensure that you have adequate resources. The backup inventory table below can assist you in calculating your annual back up requirements. By filling in the boxes and summing the results, you can create a Backup Inventory plan. Table X.X helps you consider all aspects of tape use: accident, backup, disaster restore and replacement.

Table X.X Backup Inventory

Number of backup drives		**Total number of tapes required to backup the database on each drive, referred to as a set.**		**Number of Sets in a rotation schedule**		**Annual number of rotation schedules**		**Annual number of tapes required**
	X		X		X		=	
						Add for accident replacements	=	
						Add for retirements	=	

Exchange 2000 Services and Permissions

In order to backup Microsoft Exchange 2000 Server, you must have Active Directory available and the Web Storage System services running on the server. If one of the services is not running or is unresponsive, the backup fails.

To back up data, you must have Backup Operator permissions on the local computer. Windows 2000 Backup uses the permissions of the current logon to do the backup. Third-party backup utilities can function like Windows 2000 services, which uses

permissions from the service startup parameters. These are typically the permissions set in the LocalSystem account.

Active Directory Service

While Active Directory is not a component of Exchange 2000, it is heavily relied upon. Configuration information is stored in Active Directory. User objects are kept in Active Directory and are mail-enabled for Exchange. Active Directory must be present for Disaster Recovery to function.

Internet Information Services

While Internet Information Service is not a component of Exchange 2000, it is heavily relied upon. Some Exchange configuration information, such as Internet protocol configuration and message routing information is stored on the local computer. For a successful restore, you must back up the Internet Information Service (IIS) metabase. The metabase is a structure for storing IIS configuration settings. You can view the metabase using utilities such as Metaedit and mdutil.

The metabase update service is a component in Exchange 2000 that reads data from Active Directory and transposes it into the local IIS metabase. It runs underneath the MS System Attendant Service and accepts Active Directory notifications about changes in the directory that pertain to the configuration of protocols that reside in IIS . When it receives these notifications, it updates the metabase automatically.

This service allows you to make remote configuration changes to virtual servers without a permanent connection to each system.

With NTBackup you can backup files on the local computer or on network drives, in addition to backing up the system state or exchange databases (if esebcli2.dll is installed). A filesystem backup refers to backing up files. You can select an entire hard drive and then back up the entire directory. Or you can select an individual file, and then back it up on a per file basis. If the file is locked then the backup skips the file.

Filesystem backup is not the best choice for the metabase because the metabase maintains dependencies on other components that are not saved using a straight filesystem backup. Also, the backup file might be undergoing modification at the time of backup. The best method of backup is to perform a system state backup, because this backs up the metabase. The metabase.bin file is skipped and not retrieved by a filesystem backup if IISAdmin is running.

If you only retrieve the metabase.bin file in a filesystem backup this is not enough to restore it if you need to rebuild the server. The Metabase.bin file is encrypted, and for IISAdmin to open it, the public keys for the system also need to be backed up and restored. Public Key information is in both the registry and in the \Documents and Settings\Administrator\Application Data\Microsoft\Crypto folder. It is recommended that you back up the metabase using a system state backup because of these dependencies.

Performance Considerations

You should complete backups in a timely way, and certainly before the next backup begins. For most organizations performance is measured by the length of time for the completion of the restore process. You should consider using the fastest hardware

possible. Connecting the tape drive to the server itself further speeds up the restore because the network does not slow it. Consider a tape drive that uses streaming and striping. Or, consider using a fast, wide SCSI connection. The more tape drives involved, the faster the speed in writing to backup.

If the Windows 2000 hard drive is mirrored then the transaction logs are placed on a dedicated physical hard drive, which can also be mirrored. If the Windows 2000 swap file and the Web Storage System is placed on a mirrored drive, then it can require up to twice the disk space of the database file. Therefore you shold make sure to allow sufficient disk space.

Capturing Configuration Data

To capture configuration data, it is best to perform a full backup periodically. A full backup should include the system state and all Windows and Exchange binary files. You might want to perform this backup during scheduled maintenance. A full off-line backup is not required for backing up the Web Storage System. It is recommended that you use online backups to back up the Web Storage System.

Performing Active Directory Backups

It is recommended that you also back up Active Directory information on your domain controllers. Performing a system state backup on your Domain Controllers captures Active Directory and the Sysvol directory. You should back up Active Directory and Exchange 2000 databases at the same time to avoid losing configuration or user objects on the domain controllers or Global Catalogs. It is important to back up your domain controllers because it allows you to quickly restore all your domain controllers in parallel and not have to wait for replication to update the domain controllers.

Backup Categories

The type of backup you need to perform varies depending on the importance of the data you store. Each of the backups categories has advantages and disadvantages in terms of data storage, performance and time requirements. The backups discussed are:

- Full Backup
- Copy/Daily Backup
- Incremental Backup
- Differential Backup
- Backing Up Data on a Cluster Server Node

Full Backup

A full (or normal) backup backs up the entire Web Storage System and the Exchange log files. It deletes the transaction log files with transactions that have been committed to the server database. This means that daily full backups prevent log files from monopolizing space on the hard disk. Restoring from a full backup requires only the full backup file. This is generally the preferred backup method.

Copy/Daily Backup

A copy backup is the same as a full backup except that it does not delete log files. You can create a copy backup if you want a full backup without disturbing the state of other ongoing full, incremental or differential backups. If you plan to install new software or implement a system change, you should create a copy backup.

Incremental Backup

Incremental backups back up log files prior to the checkpoint log file, and then delete them. Incremental backup backs up all transaction log files. The incremental backup deletes the transaction log files with transactions that are committed to the database. This type of backup cannot be used when circular logging is enabled. Restoring from an incremental backup requires that you have the last full backup and each subsequent incremental backup. Once this is complete, the transaction logs are applied in sequential order to the Exchange database that was restored with the full backup. It is important that all incremental backups are restored prior to starting log file replay. If not, you might lose data.

Each backup is the same size, given the same database activity. To restore the database, you need the last full backup and all subsequent incremental backups. This is the fastest backup process.

If any incremental backup is damaged (that is, if any log file in the set is damaged), then you cannot restore incremental backups made later. Remember that transactions must be replayed in order. One damaged log file prevents subsequent log files from playing again. The Extensible Storage Engine enforces this to prevent damage to the database.

Differential Backup

Differential backups back up log files prior to the checkpoint log file, but do not delete them. Each backup becomes increasingly larger in size. Differential backup backs up all transaction log files. Transaction logs are not deleted. Restoring from a differential backup requires that you have the last full backup and the most recent differential backup. This is the second fastest restore process after a full backup.

Backing Up Data on a Cluster Server Node

You should use the Exchange 2000 Backup Wizard to perform a backup of a cluster node where the cluster service is operational. On the **What to Back Up** dialog box of the Exchange 2000 Backup Wizard, select the **Back up everything on my computer** radio button.

- Be sure the node on which you perform the backup is the owner of the cluster quorum disk. To check this, stop the cluster service on all other nodes except the node running NTBackup.exe. Then choose between the following two options:
    - Select the **Back up everything on my computer** radio button to back up everything on a node. This backup includes the clustering software, cluster administrative software, the quorum, and system state.

- Select the **Only back up the System State data** radio button to back up the system state, which includes the quorum..

- To back up all cluster disks owned by a node, perform the back up from that node.

  Note During backup, NTBACKUP.EXE might report an error that says, “Completed with Skipped Files, Examining the NTBackup log, both CLUSDB and CLUSDB.LOG failed to be backed up." You should ignore this error. The quorum logs from the Cluster quorum drive are successfully backed up.

After backing up the cluster quorum disk on one node, you don't need to back up the quorum on the remaining cluster nodes. If you want, you can also back up the clustering software, cluster administrative software, system state data, and other cluster disks on the remaining cluster nodes.

Backup Process

You begin the backup process by starting the backup application. You make calls to the Web Storage System with the type of backup desired and the backup type procedure begins. The store informs the Extensible Storage Engine that a backup mode is starting. When this occurs, a patch file is generated for each database in the backup (assuming this is a full backup). If this is a differential or incremental backup then it does not create a patch file. When a backup mode starts a new log generation starts. For example, if Edb.log is the current open log file, then Edb.log is closed and is renamed to the latest generation and a new Edb.log is opened. This signals the point where the Extensible Storage Engine can truncate the logs, once the backup is complete.

When the backup begins, the agent requests the database pages to be read and sequenced from the Extensible Storage Engine. The pages are read in numeric sequence in groups of sixteen 4kb pages (though the actual size can vary) at a time. As the database engine reads the pages, they are check summed by the Extensible Storage Engine. Thus the page numbers and the checksum undergo a validity check. If they are invalid, the backup stops to prevent the storage of damaged data. Once the backup is complete and all the pages are read, then the backup copies the logs and patch files to the backup set. The log files are then truncated or deleted at the point where the new generation started at the beginning of the backup. Then the backup set closes and the Extensible Storage Engine enters normal mode and the backup is complete.

In an incremental or differential backup, only the log files are affected. Operations involving Patch files, check summing or reading pages sequentially are not executed. Figure X.X shows the Exchange 2000 backup process.

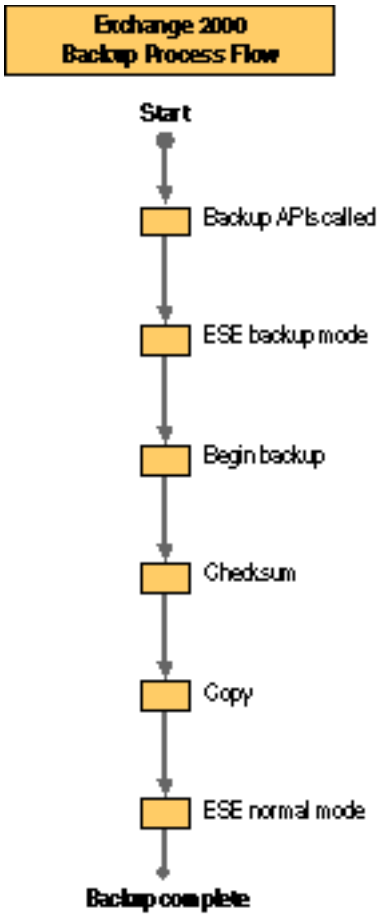

Figure X.X The Exchange 2000 Backup Process

Online Backup Process

There are two general types of backups, online and offline. It is recommended that you use the online backup type. Online backup allows the database to continue running while you are backing up data. The advantage of an online backup is that users are not affected and no processing jobs are interrupted. An online backup can be either partial or full. Full backups copy everything in the database and partial backups copy only the log files. Regardless of the category of backup performed, an online backup process always follows these steps:

1. The backup starts, a synchronization point is fixed and an empty patch file is created.
2. Edb.log is renamed to the next log number regardless of whether it is full and a new Edb.log is created.
3. Windows NtBackup.exe copies 64 KB of data at a time. Additional transactions are created and saved as normal. Each page is checked for damage.
4. Logs used during the backup process (those from the checkpoint forward) and the patch files are copied to tape.
5. Backup finishes.

Table X.X – Online Backup Types

Type	Data Backed Up	Logs Purged	Restore
Normal (full)	Database & log files	Yes	Last Normal Start LogFileReplay
Incremental	Log files only	Yes	Last Normal Every Incremental Start LogFileReplay
Differential	Log files only	No	Last Normal Last Differential Start LogFileReplay
Copy	Database & Log files	No	Last Copy Last Differential or every Incremental Start LogFileReplay

During an online backup, transactions can still be entered in databases. If a transaction causes a split operation (an internal low-level ESE operation), across the backup boundary (the location in the .edb file which designates what has been and has not been backed up), then the effected page before the boundary is recorded in the patch file. A separate .PAT file is used for each database that is backed up, such as - PRIV1.PAT, PUB1.PAT, and SRS.PAT. These files are seen only during the backup and restore processes. The steps are as follows:

1. The backup for the current storage group .EDB file begins.
2. A .PAT file is created for each database which is backed up, the database header is written into the .PAT file.
3. Split operations across the backup boundary are written into the .PAT file.
4. .PAT files are written to the backup tape with the log files.
5. .PAT files are deleted.

Offline Backup

Offline backups allow you to save a copy of a consistent database file. Offline Backups are always full backups because the database shuts down. An offline backup is always the second choice, with the online backup being preferred. It is necessary to dismount the database before performing the offline backup.

Backup Rotation Schedule

Creating and verifying daily backups is a critical step in successful data restoration. Failure to verify backups is one of the most common mistakes administrators make

because they often assume that backup tapes are being swapped and that data is being properly backed up. Make it a part of your daily routine to review all backup logs, and then investigate any errors or inconsistencies.

> **Note** The option to verify backups in NTBackup is not a recommended practice. What actually happens is when NTBackup writes data to the tape, it includes Cyclical Redundancy Checking (CRCs) within the data. Upon completion of the backup NTBackup reads the tape, verifying those checksums. The best way to verify your data is through a practice restore.

A good rotation schedule is important for data restore. The best schedule is one that provides you with a comprehensive history of file versions. Records should be stored in a secure, off-site location. Following are descriptions of two popular rotation schedules.

Month-Week-Day

The most commonly used media rotation schedule is "Month-Week-Day." This scheme uses daily, weekly, and monthly backup sets. Backup media are labeled with the day of the workweek that it backs up. Typically, incremental backups are performed on the Day group of media. This media is reused each week on the day matching its label. A set of up to five weekly backup media is labeled Week1, Week2, and so on. Full backups are recorded weekly, on the day that a Day media is not used. This Week media is reused monthly. The final set of three media is labeled Month1, Month2, and so on, according to which month of the quarter in which they are used. This Month media records full backups on the last business day of each month and is reused quarterly. Each of these media can be a single tape or a set of tapes, depending on the amount of data you need to back up. A total of 12 media sets are required for this basic rotation scheme, allowing for a history of two to three months. Because a longer history is often required, archive tapes are periodically pulled from the rotation and replaced with new tapes. Table X.X shows the Month-Week-Day rotation schedule.

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
1 Day 1 incremental	2 Day 2 incremental	3 Day 3 incremental	4 Day 4 incremental	5 Day 5 incremental	6 Week 1 normal	7
8 Day 1 incremental	9 Day 2 incremental	10 Day 3 incremental	11 Day 4 incremental	12 Day 5 incremental	13 Week 2 normal	14
15 Day 1 incremental	16 Day 2 incremental	17 Day 3 incremental	18 Day 4 incremental	19 Day 5 incremental	20 Week 3 normal	21
22 Day 1 incremental	23 Day 2 incremental	24 Day 3 incremental	25 Day 4 incremental	26 Day 5 incremental	27 Week 4 normal	28
29 Day 1 incremental	30 Day 2 incremental	31 Day 3 incremental	1 Day 4 incremental	2 Day 5 incremental	3 Month 1 Week 1 normal	4

Figure X.X The Month-Week-Day Schedule

Towers of Hanoi

The Towers of Hanoi rotation scheme is widely used. It is based on a mathematical puzzle of the same name. This schedule follows the solution to this problem. One media set, A, is used every other backup session (daily). Start day 1 with A and repeat every other backup (every other day). Set B then starts on the first non-A backup day and repeats every fourth backup session. Set C then starts on the first non-A or non-B backup day and repeats every eighth session. Set D then starts on the first non-A, non-B, or non-C backup day and repeats every sixteenth session. Set E alternates with set D.

With each additional set added to the rotation scheme, the backup history doubles. The frequently used sets have the most recent copies of the data; while less frequently used sets retain older versions. You can use this schedule either on a daily or weekly rotation scheme. You should base the decision regarding the frequency of rotation on the amount of log file transactions. To maintain the required history of file versions, you should use at least five sets in the weekly rotation schedule, or eight for a daily rotation scheme. You should periodically remove tapes from the rotation for archive purposes. Table X.X shows the Towers of Hanoi rotation schedule.

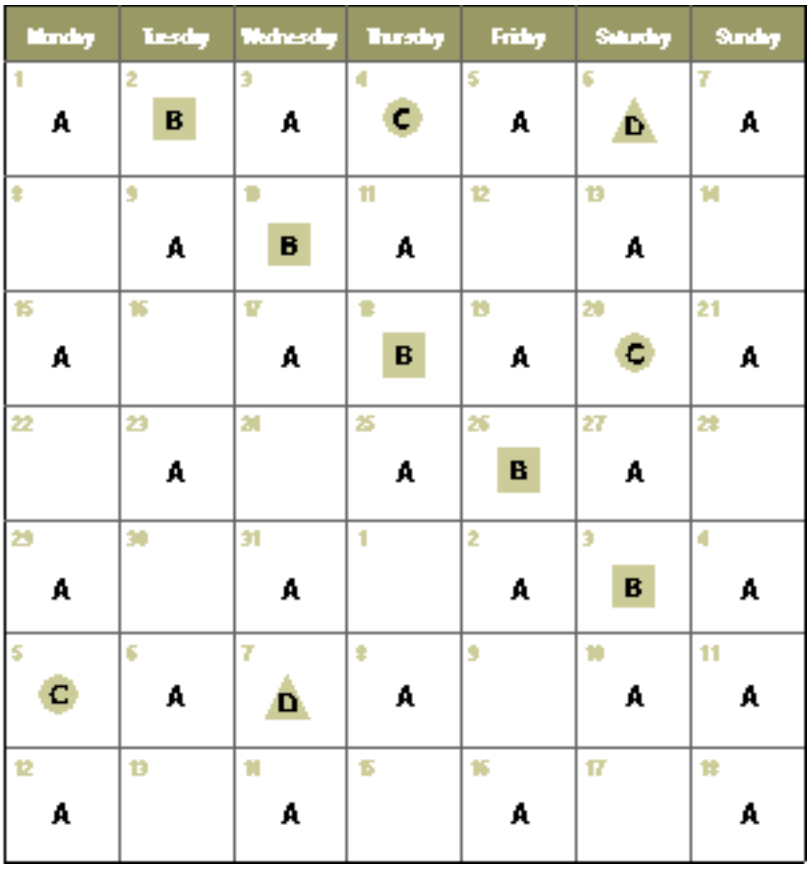

Figure X.X The Towers of Hanoi Schedule

Restore Process Overview

The database or storage group must first be dismounted and made inaccessible to users. You can do this using system manager. When a restore operation begins, the store informs the database engine and the restore mode begins. The agent (or the backup APIs) copies the database (MDB) from tape directly to the database target path The database is a file pair of the .edb and .stm files. The associated log files are copied to the server in a temporary location. This occurs because you don't want the log files saved to the same location as current log files in the exchange directory or in the production database directory. If this happens, you can over-write log files and cause a logical corruption of the database. Therefore, the restore instance of the database brings back the log and patch files to a temporary location specified by the backup operator and then restores those files to the server. After the files are restored to the drives, a new restore storage group (an instance of the Extensible Storage Engine) launches specifically for the purpose of restoring the database. Then the patch file data and the log files from the tape backup are applied to the database by the restore database engine. The current logs are then processed by the database engine, bringing you to the exact restore instance where you want to be. Once this is complete, some clean-up is done by deleting log and patch files from the temporary location and then the restore storage instance is deleted, and the storage group is mounted by the instance that actually owns it. Figure X.X shows the Exchange 2000 restore process.

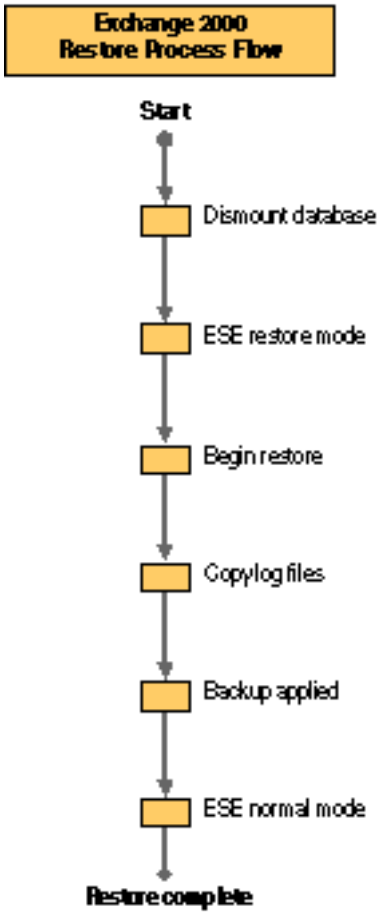

Figure X.X The Exchange 2000 Restore Process

Prerequisites for Disaster Recovery

All system functions can be restored from tape after the designated machine has been attached to the network and a suitable tape drive is available. Ensure that the IP address and domain name are assigned to the new system before proceeding. Then you may complete any vendor-prescribed pre-installation diagnostics. It is then possible to install Windows 2000 and Exchange 2000 Server. From that point on data restore from backups may take place. Data restore relies entirely upon the use of backups stored in locations off-site. Early data restore efforts focus on restoring the operating system(s) for each computer system. Then first line restore of application and user data from the backup tapes is done. The user should note that logs backed up from tape are not restored to the log directory – the user determines the location to which the logs are restored. The following are conditions you should note before restoring data:

- If the Exchange 2000 Server to which you are restoring files is a member server in a domain, be sure that Active Directory is running. Run the Exchange System Manager and verify that a valid server object still exists for the Exchange 2000 server in Active Directory. If Active Directory does not exist then restore Active Directory prior to restoring Exchange 2000.
- If the Microsoft Exchange Server you want to restore is also the Domain Controller then begin by first restoring Active Directory on that computer. You can only restore Exchange 2000 after Active Directory is successfully restored. The security ID on the restored server must match the security ID of the original server. If the security IDs do not match, you cannot access the Web Storage System until you restore only the Web Storage System and then manually rebuild the Windows 2000 accounts.

Planning Restore Strategies

The backup strategy you choose has a direct impact on what you will do for restore. In a strategy with a full daily backup, only one tape is required. Thus the volume of data and the time this procedure takes is minimized. However in a full backup with daily incrementals, you have the full backup tape from the first day of the cycle and a tape for each daily incremental backup until the point at which you wish to recover. With multiple tapes to deal with, there is a greater amount of data to manipulate and more time required to carry out the restore. Unfortunately the more tapes you have to install and the increased duration of the restore procedure increase the opportunities for you to make a mistake. The third strategy is a full daily backup with a daily differential backup. Two tapes are required, the full backup from the start of the cycle and the last differential from the day before the point at which you want to recover. Most clients use a full daily backup strategy because of its convenience and speed. Figure X.X shows three commonly used restore strategies

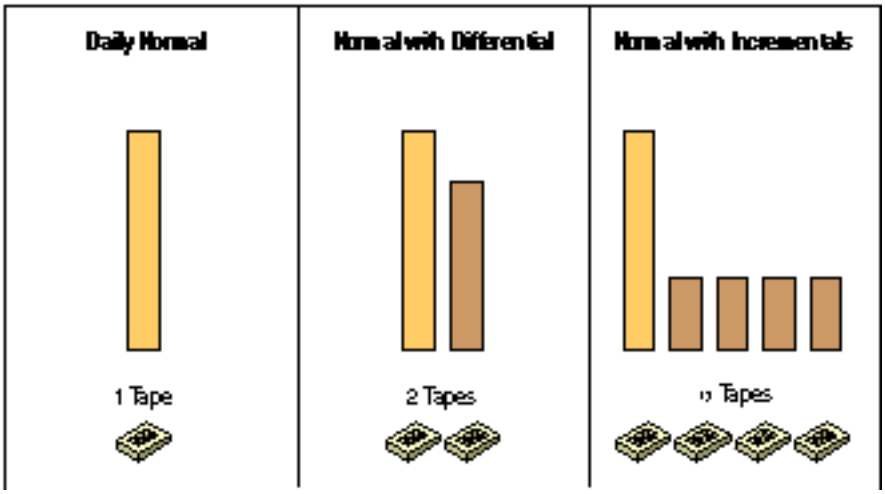

Figure X.X Restore Strategies

Determining Overall Performance

The two factors that determine overall performance are the time needed to copy files from tape to the hard drive and the time needed to replay transactions. If your system is properly configured, you should be able to reach the following speeds:

Note These numbers will vary, and customers should perform testing on their own systems.

- Backup to a single DLT 3570 tape, approximately 30 gigabytes per hour or 8.5 megabytes per second with hardware compression.
- Backup to a redundant array of independent disks RAID 5 (data striping with parity) array of four 3570 DLT tape drives, approximately 40 gigabytes per hour.
- Restoring to a RAID 5 disk partition, approximately 20-25 gigabytes per hour with write-back caching.

Note If write-back caching is disabled it takes twice as long to restore. Write-back caching should only be enabled if it it’s battery-backed.

Replaying transactions can take several hours on a large server. This is because the database must run through every transaction since the last backup. The type of transaction that must be replayed also influences speed. It takes the Web Storage System longer to replay log files with lots of deletes or attachments. Transaction replay can take, depending on the log file, a few seconds to 4 minutes per log file. Exchange 2000 plays log files significantly faster than previous versions.

Restore Categories

The type of restore you need to perform varies depending on the amount of information you need to restore. The restore process is the same for each category, however it becomes more comprehensive depending on the level of restore used. The restore categories discussed are:

- Restoring online backups
- Restoring offline backups
- Restoring log files
- Restoring a single mail box
- Restoring a single database
- Restoring Server clusters
- Full Server Restores

Restoring Online Backups

The Web Storage System databases can be restored to different server names. All transaction log files must be in place before starting the database. You cannot restore a full backup, start the database, stop it, and then restore incremental backups.

If you must rebuild a computer then you need a valid Exchange Server object in Active Directory. Use System Manager from another computer in the organization to verify that the original server name of the server you want to recover still appears in Active Directory.

Restoring Offline Backups

Restoring offline backups is more time consuming and less reliable than restoring online backups. However, the following are advantages of using an offline backup:

- You can move a database file if the hardware needs to be replaced or upgraded. Database services stop during an offline backup, so no changes take place until you restart the database.
- You can use an offline backup if an online backups fails, because you receive a -1018 error. In this event, treat your backup as a temporary backup, and take all necessary steps to correct the situation before it deteriorates further.

The following are disadvantages of using an offline backup:

- You must stop database services because offline backup copies files to tape or disk
- The database is not physically checked which allows damaged data to be copied.
- The ability to play log files created after the backup is not supported in this scenario.
- The chance of data loss is increased due to file manipulations

Restoring a Single Mailbox

If a single mailbox is completely removed, then successfully restoring it requires that you use 3rd party software or restore the entire database to a recovery server. To prevent loosing mailboxes it is recommended that you set retention times. To do this, you open the **Mailbox Store Properties** dialog box in Exchange 2000. You set the retention times on the **Limits** tab. Under **Deletion** settings, you enter a values for **Keep deleted items for __ days** and **Keep deleted mailboxes for __ days.**

If you delete a user account by accident, the user can reconnect that mailbox for a period of 30 days. This is because only the user account is removed when you delete the mailbox. The mailbox itself is retained for 30 days.

For mailboxes that are very important, you might choose to have only that mailbox, or a select few mailboxes in a single mailbox store.

If the user account is accidentally deleted, but the mailbox still exist you should reconnect the mailbox. If a user's account is deleted, his or her mailbox is marked with a red X in the **Details** pane of the Exchange System Manager. To set the Mailbox reconnection, perform the following steps:

To reconnect a deleted mailbox to a new user object:

1. From Active Directory Users and Computers, create a new user object for the user.

   Important: When creating the new user object, you must clear the **Create an Exchange Mailbox** check box. This is because you need to create a new Windows 2000 account without creating a corresponding Exchange mailbox. You connect this user account to a mailbox later.

2. From Exchange System Manager, navigate to the mailbox store where the user's mailbox is located.

3. In the **Details** pane, locate the mailbox for the user.

   Note: Verify that the mailbox icon appears with a red x. Mailboxes that display with a red X are mailboxes that have been deleted, but have been retained in the mailbox store until the deleted mailbox retention period expires.

4. Right-click the mailbox with the user's name, and then click **Reconnect**.

5. In the **New User for this Mailbox** dialog box, select the new user object you created for the user in Active Directory Users and Computers, and then click **OK.**

Restoring a Single Database

If you need to restore a single database, you should dismount it from the Web Storage System and then use NTBackup to restore the data. An extra storage group is dynamically created for the restore process and the database and transaction log files from the backup are restored into that process. The transaction logs are applied to the database in the restore storage group. After recovery, the consistent database is mounted into its original storage group by the web storage system.

To restore a database to a different server, the database display name and the storage group display name must be the same. However, the organization name and administrative group name for the server to which you want to restore must match the server from which the database was backed up.

> **Note** The format of transaction log files is revised in Exchange 2000. When you upgrade from Exchange 5.5 to Exchange 2000 the existing transaction log files are removed and a new log series is created. Because of the log format change, you cannot restore Exchange 5.5 databases onto an Exchange 2000 server.

Moving a Database to a Different Storage Group

When using Exchange 2000, you might want to move a database to a different storage group. To perform this move, you should perform the following steps

To move a database to a different storage group:

1. Open the **Mailbox Store Properties** dialog page.
2. Select the **Database** tab.
3. Select the **This database can be overwritten by a restore** check box.

This allows the database to move into a different storage group by a copy or a restore. The GUID in the database is changed to match the GUID in Active Directory.

Restore Constraints

Restoring data in a consistent way requires that you consider many different types of circumstances. Of particular interest to the network administrator is restoring Exchange data to a non-production server.

Restoring the Web Storage System to a Different Server

You can restore the Web Storage System to a Microsoft Exchange 2000 Server other than the one from which it was backed up. You use this method as a last resort to restore individual items, such as messages or folders, to a server other than the one in use. The additional server must meet the hardware requirements to run Microsoft Exchange 2000 Server. The additional server should not be connected to the network and should have enough disk space to restore the entire backup.

You should not use this method when a server fails because this method restores only the Web Storage System; it does not restore the directory. When you finish restoring the backup, you should reconfigure permissions on the mailboxes.

Restoring Exchange Data to a Non-production Server

You restore Exchange Data to a non-production server to perform a single mailbox restore when the normal retention time has expired. You rarely need to perform this type of restore because Exchange 2000 has other features such as dumpster and mailbox retention that make mailbox recovery simple. To restore data to a non-production server you need to fulfill the following requirements:

- The non-production restore server should reside on a separate Windows 2000 domain forest from the production network and servers
- The storage group that hosts the restored database must have the same display name as the original production server.
- The database you want to restore must have the same display name as the original production server.

- The database name must be unique on the backup server in all storage groups. For example if the MDB name is PRIV.edb, there can only be one instance of a PRIV.edb database on the secondary server.
- Need the same organization name and administrative group name.
- To recover the mailbox, reconnect the mailbox as described earlier in the section, **Restoring a Single Mailbox**. Use Outlook to connect the mailbox and download to the client computer.

Database Consistency

During normal operations of the Extensible Storage Engine database, the database file is never completely up-to-date with changes made in memory. Although the database file and the memory pages make the Extensible Storage Engine database consistent, the database files themselves are inconsistent. Consistency of a database file is flagged in the database file header, so that if the database is ever stopped improperly that flag tells you that when the database service restarts, the Extensible Storage Engine will replay transactions and possibly rollback any incomplete transactions. When a database service shuts down properly, all transactions are written from memory to the transaction file. If no errors occur, then the database closes in a consistent state and the flag is set to consistent.

If a database file is inconsistent, treat the contents of all database folders (mdbdata) as if they were a single file. Move them together and back them up together. Never move, rename, replace or delete files in the database folders just to see what happens. Don’t alter a file unless you know exactly why the database doesn't need it during startup. The following items should be considered when attempting to achieve database consistency:

- Defragmenting Databases
- Soft Recovery
- Log Transaction Redo
- Maintaining Database Integrity
- Using ESEUTIL
- Using ISINTEG

Defragmenting Databases

Defragmenting is the moving of data to fill database pages more efficiently. Defragmenting makes used storage contiguous and eliminates unused storage. There are two types of defragmenting, online and offline.

Online Defragmenting

Online defragmenting runs as part of database maintenance (this automatically detects and destroy objects that are no longer being used, thus freeing up database space), and does not decrease the size of the database. Online defragmenting is performed automatically at 2:00 AM everyday by default. You can change this by altering settings in Database maintenance, on the Exchange Database Properties page.

Offline Defragmenting

Offline defragmentation creates a new database copying all records and tables from the old database into the new database. Since this is a copy, defragmentation requires free disk space equal to the size of the database. Upon completion, ESE considers the new database to be a totally different database from the original. Therefore the original database is deleted and it's member log files cannot be replayed into the successor database. A full backup should be completed as soon as possible.

> **Caution** You should only defragment if a large number of users are moved off of a server, which frees up a large amount of memory. Offline defragmenting is not a recommended regular maintenance routine.

Database Size After Defragmentation

Although an Extensible Storage Engine database can perform an online defragmentation, it does not reduce the physical size of the database. At times you can purge a large number of email messages to free memory. However, the database file size generally grows at a steady rate to a point at which the amount of data removed matches the amount of data added.

An offline defragmentation reduces the physical size of a database by writing data from the used pages to a new database created by the ESEUTIL. This utility defragments, repairs and checks the integrity of the Exchange Web Storage System. When ESEUTIL starts, it creates a new database and copies database records to it. It discards unused pages. The result is a compact database file.

You should not perform a defragmentation across the network. By default, ESEUTIL prefers the contents of a live database to a temporary one. When you use the command, TEMPDFRG.edb and TEMPDFRG.stm, the temporary new database replaces the original. There are switches to control where the temporary database is created and whether the original database file is deleted. It is not recommended to locate the temporary files across the network. This will cause defrag to take longer and will slow down the network during that time. For more information about ESEUTIL, see "Using ESEUSTIL" later in this chapter.

After a defragmentation or repair is run, new database signatures are assigned. Because this is a new database, the database signature is changed; therefore previous transaction logs can no longer be replicated into this database.

Soft Recovery

In a Soft Recovery, a database starts normally and the storage group is initialized. Normally the database file is in a consistent state, so the Extensible Storage Engine simply begins to handle transactions. If the database is inconsistent, that is, it was not shut down properly, then the Extensible Storage Engine replays transactions from the checkpoint through to the Edb.log. This file must be present, if the checkpoint file doesn't exist, then the Extensible Storage Engine starts with the earliest transaction log it finds and replays transactions. In this situation, you see events logged and the application event log shows log files being replayed. When the Extensible Storage Engine finishes replaying the transaction, the database starts.

Log Transaction Redo

The Extensible Storage Engine reads an operation, which includes the page that was changed. The Extensible Storage Engine works through every operation in the transaction log files through to the last log file, which might or might not be Edb.log. If the Extensible Storage Engine finishes a series of operations, and there is no commit operation for that transaction within the log file, then the Extensible Storage Engine reverses the other operations that make up the transaction. This maintains the integrity of the database.

Using ESEUTIL

The Extensible Storage Engine Utility (ESEUTIL) is a database level utility that is not application specific. ESEUTIL extracts or dumps header information from database, checkpoint and transaction log files to provide diagnostic information. The ability to dump file header information is especially useful when you want to check the current state of the database and transaction log files. You can use this information to discern why the database might not start, what transaction logs you need to replay or to which log file the Edb.chk file points.

File Header Output

By dumping file header information, there are several output lines that you can actually pull information from. One is DBSignature; this is used to match the database file (*.edb), stream file (*.stm), and transaction logs. The signatures used by ESE are created with a random number and the date stamp of when the signature was created. Another output line is State. This specifies whether a database is consistently shut down or not. A database restored from an online backup is always inconsistent whereas a database restored from an offline backup should always be consistent.

The Output of eseutil /mh will include the following pieces of detailed information, for example:

- DB Signature: Create time:03/30/2000 18:17:00 Rand:68947200 Computer:
- cbDbPage: 4096
- State: Consistent
- Scrub Date: 00/00/1900 00:00:00
- Repair Date: 00/00/1900 00:00:00
- Last Consistent: (0x16,F,159) 03/31/2000 12:11:46
- Last Attach: (0x1,1404,17A) 03/31/2000 11:49:44
- Last Detach: (0x16,F,159) 03/31/2000 12:11:46
- Log Signature: Create time:03/30/2000 18:16:56 Rand:68924870 Computer:
- OS Version: (5.0.2195 SP 0)
- Previous Full Backup:
- Current Incremental Backup:

Transaction Log File Headers

To extract transaction log file header information, use ESEUTIL/ml and the transaction log. ESEUTIL/ml helps you determine why transaction logs don't play back into the database. For example, if you type ESEUTIL/ml edb00012.log, the output is the following parameters:

- **L-generation** log generation number in decimal
- **Signature** transaction log file signature
- **Database location** hard-coded into this file and cannot be changed except through the restore process
- **last-attach** references the log file that was Edb.log the last time the Extensible Storage Engine started

These parameters produce three numbers. For example, if the number is: XXXYYYZZZ, then XXX is a transactional log file number, in decimal from; YYY is a page within the file and ZZZ is the byte set within the page.

> **Note** The term, Last Consistent, refers to the log file that was Edb.log at the last stop of the database. This gives the same three numbers as the last attach.

Repairing Damaged Databases

The repair function of ESEUTIL examines the structure of the database tables and records broken links. The Extensible Storage Engine repair mode attempts to restore links but the process is slow and data loss is highly likely. You should only use this as a last resort. Running the repair function ESEUTIL/p attempts to restore the damaged database. The output looks somewhat like ESEUTIL/g, but the database is read/write.

You might encounter physical damage to a database, where you receive -1018, -1019 or –1022 errors, the database page is removed, and data is lost. Any repair rewrites the database signature. You should also perform a full back up of the database. If you restore a backup made before the repair, then the database is rolled forward to the state it was in at the time of the repair. You cannot replay transaction logs that are created after the repair. You should never try to repair a directory database.

Checking Database Integrity

You can use ESEUTIL/g to check database integrity. This is a read-only utility that does not make changes to the database. It checks the database tables, rebuilds all indexes on a temporary database (Tempedb.edb), and compares the indexes. There are several switches to control the output and number of logs this command creates.

Forcing Soft Recovery

This unique property allows an administrator to deal with a potential disaster. For example, a server has three databases, and experiences a hard drive failure. One of the databases was located on the lost hard drive. The server will first attempt a soft recovery. Once it encounters the lost database, it will report that the databases are no longer consistent and will cease the soft recovery. To get around this problem, the administrator can run ESEUTIL/R/I. R stands for **soft recovery** and causes the server to resume the soft recovery. I stands for **ignore hanging attach records**, and causes the

server to ignore the missing database. After log redo completes, the first two databases will be consistent. The missing database will have to be restored from backup.

USING ISINTEG

Information Store Integrity Checker (Isinteg) is a built-in component of Exchange 2000. Isinteg is a suite of tests that check the database for inconsistencies. You should dismount the mail store or public store before you run Isinteg. You should be selective about which tests you perform. Running the full complement of tests on a 40 gigabyte database can take 12 hours to complete. Figure X.X shows the basic Database Schema

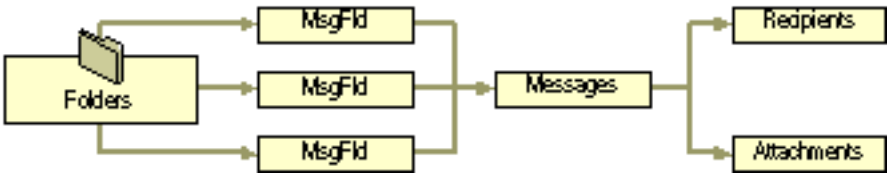

Figure X.X Basic Database Schema

Each Message Field (MsgFld) table connects a folder to its messages. It also contains per recipient message properties and duplicates of message properties that have been requested for browsing. A MsgFld table is created for every folder in the Folders table the first time a message is created in that folder.

The Messages table contains all per instance data for the message. It includes links to the Recipient tables and the Attachment tables. Messages can share recipient and attachment descriptions. There is one Messages table for the entire database.

The Attachments table contains all per-instance data for the attachment. There is one Attachments table for the entire database. The Attachments table is also a message folder table. It has a row in the Folders table, and the display name is MDB Attachments.

The Recipient table contains all per instance data for a recipient. There is one Recipient table for the entire database.

> **Note** If a storage group already has 6 databases, then ISINTEG will fail because it must be able to create a temporary database in order to run. Therefore only run ISINTEG with a maximum of five databases in the storage group. Customers always have to dismount a database before running ISINTEG. So if they have 6 databases in a storage group, and they want to run ISINTEG, then In addition to dismounting the database they want to run against they also have to dismount another database. Then ISINTEG can create a temporary reference database and everything will work fine.

Reference Database And Multiple Passes

Isinteg uses a reference database to check reference counts. It operates in three passes.

- **Pass 1**, Constructing the reference database: In the first pass, it searches all of the original database, and constructs a reference database that has many tables. Each

table has as many rows as there are in the original database. But each table has only one column, the key. And the key is copied from the original database. The test is labeled reference table construction, in Isinteg's output.

- **Pass 2**, Filling the reference database: In the second pass, the reference database is filled when the different tables in the original database are searched. This can be done in more than one test. In this pass, random searches into the reference database tables occur.
- **Pass 3**, Cross-checking the original database columns with the reference database columns: In the third pass, every table in the reference database is searched sequentially, along with the corresponding table in the original database. Each column in the reference database is compared with the corresponding column in the original database, and any discrepancies are reported (and corrected if in the -fix mode). In Isinteg's output the test is labeled Reference Table Verification.

Running Isinteg

To run Isinteg, you must type the following at the command prompt:

```
isinteg -s [-fix] [-verbose] [-l logfilename] [–t refdblocation] -test
testname[[, testname]...]
```

Table x.x shows the meaning of the terms you use to run Isinteg

Term used on command line	Meaning
-s	Server Name
-fix	Specifies fix mode. Isinteg will fix the problems it detects, if in this mode. The default is check only mode, where problems are reported but are not fixed
-verbose	Report verbosely
-l logfilename	Specifies the name of the output log file. By default, the results of check/fix operations are dumped to isinteg.pri or isinteg.pub as the case may be, in the directory from which isinteg was executed.
-t refdblocation	Location of the temporary database (the reference database) that Isinteg constructs during execution. This will be removed automatically by Isinteg when it terminates. Specifying the location of the temporary database on a different disk may improve performance because it facilitates parallel access.
-test testname	The Isinteg test selected

Table x.x Terms you use to run Isinteg

Note For more information about the tests that Isinteg performs, see the appendix, "Isinteg Test Suite Table" in this book.

Recovering from Disasters

We define a disaster as having to restore Windows 2000 Exchange Server and/or Windows 2000 Server. With Exchange 2000 the introduction of multiple storage groups and databases adds more complexity to restoring. But in addition backup and restore with Ntbackup, reinstalling Exchange 2000 is now run with the /DisasterRecovery option. This option allows you to run setup in Disaster Recovery mode in order to rebuild a server previously lost in the Exchange topology.

Besides Disaster Recovery setup, you must be aware of the different procedures needed for recovering servers. With Exchange 2000 servers now take on different roles such as Key Management Server or Site Replication Server. In addition to Exchange server roles, the platform Exchange is running on also adds more steps. Recovering an Exchange 2000 cluster server requires more steps than recovering a single Exchange 2000 member server.

Requirements for recovery and four server scenarios and their steps for recovery are described in this section:

- Requirements for Recovering Exchange 2000
- Recovering an Exchange 2000 Member Server.
- Recovering an Exchange 2000 Member Server Running Site Replication Service (SRS)
- Recovering an Exchange 2000 Member Server Running Key Management Service (KMS**)**
- Recovering an Exchange 2000 Cluster Server.

Requirements for Recovering Exchange 2000

There are five common requirements for recovering all Exchange 2000 servers.

- **Windows 2000 and Exchange 2000 Installation CDs.** This includes any applicable service packs or hot fixes as outlined in the Windows 2000 and Exchange 2000 release notes.
- **Full backups of the system drive.** This includes any other logical drives where critical application data is installed.
- **Recent Windows 2000 System State backup.** A System State backup is a an important type of Windows 2000 backup. Windows 2000 NTBackup captures and saves system configuration information that a filesystem backup normally fails to back up. System configuration information such as the Windows registry and IIS metabase data are included in a System State backup.
- **Backups of Exchange databases.** These are online backups of the Web Storage System databases.
- **Active Directory must contain a server object for the server you want to restore.** If an Exchange 2000 server is damaged it is important that the server

object in Active Directory is intact. If the server object is deleted from Active Directory, either using the Exchange System Manager snap-in or Active Directory Users and Computers snap-in, then recovery cannot succeed. Following a disaster, you must avoid changing the server object until the actual server is recovered.

Important: These steps assume that your Exchange 2000 server is a member server in a domain and not the server that runs Active Directory.

If the Exchange 2000 server is also an Active Directory server be sure to include backups of Active Directory in the System State backup. For information about the requirements for restoring Active Directory on a Domain Controller, see "Backup" in the Windows 2000 Server Operations Guide.

Recovering an Exchange 2000 Member Server

If Active Directory runs on a separate Domain Controller in the domain and it is intact, proceed with the steps below. If Active Directory is running on the same computer as the Exchange 2000 member server, you must first restore Active Directory before restoring Exchange 2000. For more information about restoring Active Directory see "Backup" in the Windows 2000 Server Operations Guide.

In addition to reinstalling Windows 2000 on the computer and restoring filesystem backups, it is recommended that you restore the Windows 2000 System State. When you complete a backup of the Windows 2000 System State, which includes the Windows registry and IIS metabase, the computer returns to the state it was in before the backup.

Reinstalling Windows 2000

You might need to reinstall Windows 2000. If so the steps for reinstalling it are as follows:

1. Install Windows 2000 as a stand-alone server.
2. Install the same version of Windows 2000 that was previously installed. For example, you might have Advanced Server or Data Center Server installed.
3. Install Windows 2000 to the same hard drive and paths to which it was previously installed. If you need to, configure the drives to match the previous logical drive configuration.
4. Use the same server name as the original server.
5. Select all of the same components installed on the original server.
6. Do not rejoin the domain. Leave the server in a workgroup. You should do this because, after installing Windows 2000, you should restore the System State that places the server in the correct domain.

Important: After reinstalling the correct version of Windows 2000, re-install any service packs or hot fixes that were previously installed.

Restoring the System Drive

You should restore full backups of the system drive or any other logical drives where critical application data was installed. You should use NTBackup to restore filesystem backups to the computer on which you restored Windows 2000.

Restoring Windows 2000 System State

You should restore the Windows 2000 System State. By restoring the Windows 2000 System State the restored computer returns to its original domain where its computer account matches the System ID (SID) in Active Directory. You should use Ntbackup to restore the Windows 2000 System State. After the restore, Ntbackup prompts you for the computer you want to restart.

> **Important** After restoring the system state and restarting the server, you might see an error dialog box saying that one or more services cannot be started. This includes services such as SMTP that were running prior to restoring System State, and services that have not been installed yet. These services require that Exchange 2000 be installed. If the full filesystem restore does not include the Exchange 2000 installation directory or other critical application data you see this error. These services start after Exchange 2000 installs in disaster recovery mode.

Following the restore of your System State, the event log might show that some Exchange 2000 services have failed. If these services are not installed yet, when you restore the System State, Windows 2000 accepts that these services are installed on your server. These services start after Exchange 2000 installs in disaster recovery mode.

Run Exchange Setup in Disaster Recovery Mode

When you run Setup.exe with the /DisasterRecovery option, Exchange 2000 restores executable files and system settings without disturbing the existing Active Directory information for the system. Setup in Disaster Recovery mode installs Exchange 2000 without resetting the server's configuration to defaults but instead leaves the server in its last configuration.

To run Exchange Setup in disaster recovery mode:

1. From your Exchange 2000 CD run *setup /DisasterRecovery*.
2. In the Exchange 2000 Setup Wizard be sure every component originally installed on the computer is set to the Disaster Recovery action. If the originally installed components are not selected for Disaster Recovery then manually select them.

   > **Important:** You must install Exchange 2000 to the same drive and directory on which it was installed on the original server.

3. During disaster recovery, a dialog informs you that you cannot restore Exchange 2000 unless Active Directory still contains a server object for the server being restored. Use Exchange System Manager to verify that the server object exists for the server you are restoring. If the server object does not exist, the recovery process does not succeed.
4. Near the end of the setup in Disaster Recovery mode you are instructed to restore databases at the end of the setup and then restart. If setup completes and displays

another dialog box that prompts you to restart, you should ignore this message and restore the databases before you restart.

Recovering Databases

To recover databases in the Web Storage System, you must verify that all services on which Exchange 2000 depends are running. To restore a database, you must also dismount the database. However, because Exchange supports multiple storage groups, you only need to dismount the specific database you want to restore. This allows users access to any other databases in the Web Storage System.

To recover databases in the Web Storage System:

1. Use NTBackup to restore your databases. In the **Restoring Database Store** dialog box's, **Temporary location** text box, specify a directory in which to store a log file that is different from the directory where the original log files exist. Make sure the location has enough disk space to store the files. If you restore databases or log files to their original location, any existing databases or log files are overwritten.
2. If you are restoring a full backup without any incremental backups, select **Last restore set** to start the log file after restoring the database. If you are restoring a backup with incremental backups, do not select this option until you restore the last incremental backup.
3. After you finish the restore, verify that the databases are mounted, before you restart the system. The Exchange 2000 member server restores after you restart the system. Configuration settings that existed before the original server was damaged should remain. The restored server can return to its previous role in the Exchange organization.

Recovering an Exchange 2000 Member Server Running Site Replication Service

Recovering an Exchange 2000 member server also running Site Replication Service (SRS) requires the same steps involved in recovering a single member server. However there are additional steps to recover the SRS database following Disaster Recovery setup.

Besides the requirements described in "Requirements for Recovering Exchange 2000" earlier in this chapter, you should also back up the Exchange 2000 Site Replication Service (SRS) database. Follow the same steps listed in "Recovering an Exchange 2000 Member Server" earlier in this chapter. Reinstall Windows 2000, restore the system drive backup, restore the Windows 2000 System State backup, run Exchange 2000 setup in Disaster Recovery mode, and restore Exchange Web Storage System databases. However, prior to restarting the computer as described in the last steps of "Recovering Databases" section of "Recovering an Exchange 2000 Member Server" listed earlier in this chapter, you should, perform the steps listed below to recover the SRS database from backup. Recover the SRS database using Ntbackup. Then restart the server.

To enable and start SRS following disaster recovery:

1. Using the Computer Management snap-in select **Services** under **Services and Application.**
2. Select **Exchange Site Replication Service** from the list of services and then click **Properties**.
3. For the **Exchange Site Replication Service** set the startup to **automatic** then start the service

To reset the password for the Exchange 5.5 service account:

1. Using the Exchange System Management snap-in select **Administrative Groups** and select your site name.
2. Right-click the site name and then click **Modify**. Clear the password and confirm password fields. Re-enter both the password and confirm the password fields.
3. Click **Apply**.

Restoring an SRS Database

Using Ntbackup, follow the same steps listed in "Recovering an Exchange 2000 Member Server" earlier in this chapter. However, instead of selecting **Exchange Information Stores with Ntbackup**, select the **SRS database** to be restored.

When you restore the SRS database, you have completed the recovery of the Exchange 2000 member server running SRS. Verify that all the steps described in "Recovering an Exchange 2000 Member Server" earlier in this chapter, have been performed before you restart the system.

Recovering an Exchange 2000 Member Server Running Key Management Server (KMS)

Recovering an Exchange 2000 member server also running Key Management Service (KMS) requires the same steps involved in recovering a single member server. However there are additional steps to recover the Certificate Authority (if Certificate Authority was running on the same server as KMS) and KMS database following Disaster Recovery setup.

Besides the requirements described in "Requirements for Recovering Exchange 2000" earlier in this chapter, you should be sure to have the following backups as well:

- Backup of Exchange 2000 Key Management Service (KMS) database.
- System State backup containing Certificate Authority if KMS server also runs Certificate Authority.

Follow the same steps listed in "Recovering an Exchange 2000 Member Server" earlier in this chapter. Reinstall Windows 2000, restore the system drive backup, restore the Windows 2000 System State backup, run Exchange 2000 setup in Disaster Recovery mode, and restore Exchange Information Store databases.

If the recovered server was also the Certificate Authority, in addition to being the Key Management Server; then by restoring the system state of the original server, Certificate Authority is also restored. Certificate Authority does not have to be checked as a component to install if you are restoring the System State.

Prior to restarting the computer as described in the last steps of "Recovering Databases" in the "Recovering an Exchange 2000 Member Server" section earlier in this chapter, you should perform these added steps in recovering the KMS database from backup. Then restore the KMS database using NTBackup reboot server.

To start KMS following disaster recovery:

1. Using the Computer Management snap-in, select **Services** under **Services and Application.**
2. Select **Exchange Key Management Service** from the list of services and select **Properties**.
3. For the **Key Management Service** enter the Key Management password in **Startup Parameters** and select **Start.**

Restoring KMS Database

Using NTBackup perform the steps as described in "Recovering Databases" in "Recovering an Exchange 2000 Member Server" earlier in this chapter. However, instead of selecting **Exchange Information Stores** with NTBackup, select the KMS database you want to restore.

The Exchange 2000 member server running KMS recovery is complete after you restore the KMS database. Verify that the steps described in "Recovering an Exchange 2000 Member Server" earlier in this chapter are complete and then restart the system.

Recovering an Exchange 2000 Cluster Server

Clustering provides a mechanism for moving resources between cluster nodes when a disaster occurs. In the case where a single server fails, clustering will move Exchange 2000 resources to another server in the cluster so services remain available to users. The server that failed can be removed from the cluster and then later replaced with another server joining the cluster. Exchange resources can then be moved back to this newly joined server so that load balancing is again achieved. The steps involved in removing a non-functioning server from a cluster, rebuilding, and then rejoining the server to that cluster, are described in this section.

In addition to disasters involving loss of single nodes in a cluster, there are cases where the cluster-shared disk is lost. This section describes how to recover when the cluster quorum is lost.

Recovering A Single Server In A Cluster

Clustering provides recovery for when a server node goes down in a cluster. When a single server in a cluster fails, Exchange resources running on the server are moved to an available node in the cluster. Exchange databases remain intact on shared storage and can be accessed by the Exchange Virtual Server from another node in the cluster. This is a feature of clustering, and provides reliability when disaster occurs on a single server in the cluster. Once resources are moved to an available node in the cluster, follow these procedures for removing the non-functioning node and replacing it with a new node.

Evicting The Lost Server Node From The Cluster

When a server suffers a disaster in a cluster and needs to be replaced by a new node perform these steps.

To evict the lost server node from the cluster:

1. Use Cluster Administrator to remove the lost node from the cluster.
2. Using Cluster Administrator, verify that the evicted node for each cluster group and resource does not appear as a possible or preferred owner.
3. Physically remove the damaged server from the cluster and shared storage.

Rebuilding the Server Node for Cluster

The lost node does not have to be rebuilt like the original lost node. An entirely new node can be built (new machine name, IP) and then joined to the cluster. Perform the following steps to create a new server node.

To rebuild the server node for the cluster:

1. Install Windows 2000 on the computer and provide a new computer name.
2. Join the same domain as before with same administrative permissions given to the Exchange Administrator account as before.
3. Setup the new computer to access the same shared storage as the original node.

Rejoining the Server Node to the Cluster

To rejoin the server node to the cluster, you should setup cluster service on the newly built server. When asked to join a cluster you should specify the cluster you want to join.

Install Exchange on the Server Node and Move Resources Back to the Node

You might need to install exchange on the server node and move resources back to the node.

To install exchange on the server node and move resources back to the node:

1. Install Exchange 2000 on the newly joined node. Before Exchange resources can be moved back to the newly joined node Exchange must be installed.
2. Verify that the cluster groups and resources on the other node show that the new node is a possible or preferred owner.
3. Move the Exchange resources that originally failed over back to the new node.

Recovering A Lost Cluster Quorum

In order to recover from a cluster quorum failure, you need to perform a cluster quorum backup. In addition to the requirements described in "Requirements for Recovering Exchange 2000" listed earlier in this chapter, you should have a System State backup containing the cluster quorum as well:

Typically you need to recover a single lost node in a cluster. However, you might have a case where the cluster quorum is lost on the shared disk along with the Exchange databases. If this occurs, you should perform the following steps.

Restoring the Cluster Quorum From Backup

Before you can restart cluster service on any nodes in the cluster, you need to restore the quorum.

To restore the quorum:

1. Use the Windows 2000 Resource Kit Dumpconfig utility to restore the signature of the quorum disk if the signature of the quorum disk has changed since you backed up.
2. If the Cluster service is running, stop the Cluster service on all cluster nodes.
3. Restore the system state (containing the contents of the cluster quorum disk) using Ntbackup.exe. Ntbackup.exe puts the contents of the cluster quorum disk in subdirectory *systemroot*\cluster\cluster_backup.
4. After restoring you are prompted to restart. Instead of restarting, run Clusrest.exe to restore the content of the *systemroot*\cluster\cluster_backup directory to the cluster quorum disk. The clusrest.exe tool can be found in the Windows 2000 Resource Kit.
5. Restart the computer.

Restoring Exchange 2000 databases From Backup

Following quorum restore and restarting of the nodes in the cluster, verify whether the shared disk resource can be accessed after the cluster service has started. If the shared disk where Exchange databases reside can be accessed and has survived the disaster, check to see if edb, stm and log files still exist for the Exchange Virtual Server storage groups. If intact proceed to start your Exchange resources. If the shared drive is lost then perform these steps to restore your Exchange databases from backup.

To restore Exchange 2000 databases from backup:

1. Start Exchange System Manager and select the **Do not mount at startup** check box for databases owned by the Exchange Virtual Servers on the cluster. This is avoids creating new databases on the shared disk resource when the Exchange resources start.
2. Using Ntbackup, perform the steps as described in "Recovering Databases" in "Recovering an Exchange 2000 Member Server" earlier in this chapter.

   Note On a cluster server you should verify that the shares where exchange databases reside are available, and accessible, by the cluster node that is the owner of the disk resource.
3. Verify that databases are mounted with Exchange System Manager and check the Event log. Be sure to de-select the **Do not mount at startup** check box for each database that is successfully restored.

Best Practices

There are many recommended procedures for achieving successful backups and restores. The following sections outline some of the best practices for specific backup and restore situations.

Increase Backup Tape Reliability

You should note the capacity of you backup tapes in relation to the size of the database. Ensure that the raw storage capacity of your tape exceeds the compressed storage capacity of your database by a comfortable safety margin. If it does not, plan for tape changes when doing backups. For example, you must change a tape with a raw capacity of 4 GB twice if you use it to back up an uncompressed database of 10 gigabytes, assuming the data is compressed during the backup. You must be careful because all DLT (digital linear tape) drives use the LZW (Lempel-Ziv-Welsh) compression algorithm. Compressed data or random data (such as encrypted data) can actually expand by about five percent if you attempt to compress it further, which can reduce drive throughput.

> **Note** The LZW compression algorithm is an industry standard that reduces the number of bits to transfer without losing data. Transmitting data is compressed at a ratio of 2.8:1 at 9.6 Kbps. This is equivalent to transmitting non-compressed data at 27 Kbps. Such numbers reflect an ideal situation, and a ratio of 1.5-1.8:1 is the normal experience. The switches with this feature can transmit in either compressed or non-compressed mode.

To increase reliability, you should take the following precautions:

- Routinely clean the tape drives according to manufacturer specifications.
- Do not over use tapes. Discard them once they reach the maximum number of cycles as specified by the manufacturer.
- Store the tapes in a safe and accessible location.

Verify and Validate Backups

The ability to restore data and servers depends on the quality of your backups. Therefore, it is important to verify the success of your backup procedure. For complete fault tolerance, you should perform verification at two levels. You should verify the event and verify the data.

> **Note** If you turn on detailed logging for NTBackup, the reports contain more detailed data about what is actually backed up.

Verifying the event

It is important to verify that the backup occurs without errors. Examine the Windows 2000 Backup log for the backup event you want to verify. Thoroughly review the events in the Windows 2000 event log to ensure your backup completed as scheduled without errors. You should research and resolve errors or inconsistencies in the logs as soon as possible.

Verifying data

Verifying data includes checking the data from the storage device to another computer to verify that the backup works correctly. Verifying all backups from all servers in a large installation is difficult. However, by rotating a restore process to include various servers or backup devices, you can verify the integrity of the system and identify

potential problems before you lose data. This process also helps train new administrators to perform restore procedures.

Document and Archive Backups

It is important to document your backup strategy and provide step-by-step instructions that describe how to use backups to restore data. Backup and disaster restore procedures should also specify escalation procedures and the members of the organization responsible for approving the escalation of the situation if any part of the process fails. Label backup media and store them in a secure location. For maximum reliability, archive full backups at a separate location, preferably offsite.

Test Backups

To prepare for server failures, practice your backups, using different scenarios. It is much easier to deal with an emergency when you are prepared. Run your backups on a test server, duplicating your working conditions on a weekly or monthly basis until you are proficient. Be sure that you can deal with the following three scenarios:

- **Full Restore of Windows 2000** This can happen if a Service Pack or patch is not installed correctly.
- **Full Restore of Microsoft Exchange 2000** This can happen if there is systemic or database damage.
- **Full Restore of Microsoft Exchange 2000 and Windows 2000** This can happen if there is a power outage.

Note Do not assume that a database is damaged. The only error that can be relied on 100% of the time to prove damage is the repeatable error message -1018. If you assume incorrectly that a database is damaged this might lead you to take drastic measures that can prolong downtime and cause unnecessary data loss. Most startup failures that are blamed on damage are really caused by the presence of mismatched files or the absence of necessary files.